Arch. Math. Logic 44, 167–193 (2005)
Digital Object Identifier (DOI):
10.1007/s00153-004-0242-2

Mathematical Logic

Roberto Maieli · Quintijn Puite

Modularity of proof-nets

Generating the type of a module

Received: 8 May 2003 / Revised version: 15 December 2003 /
Published online: 25 May 2004 – © Springer-Verlag 2004

Abstract. When we cut a multiplicative proof-net of linear logic in two parts we get two modules with a certain border. We call pretype of a module the set of partitions over its border induced by Danos-Regnier switchings. The type of a module is then defined as the double orthogonal of its pretype. This is an optimal notion describing the behaviour of a module: two modules behave in the same way precisely if they have the same type.

In this paper we define a procedure which allows to characterize (and calculate) the type of a module only exploiting its intrinsic geometrical properties and without any explicit mention to the notion of orthogonality. This procedure is simply based on elementary graph rewriting steps, corresponding to the associativity, commutativity and weak-distributivity of the multiplicative connectives of linear logic.

1. Introduction

> "Switchings should be seen as dense subset of para-proofs."
> [J.-Y. Girard, On the meaning of logical rules I, 1998, p. 40–41]

The notion of modularity plays a central role in the current programming languages and techniques: the major part of them make sensitive use of concepts like *modules, objects, components*, etc. When we build a large program it is natural to cut it into reusable sub-units, or modules, which should result to be correct; this correctness must be checked locally, without involving the whole program. Moreover, when we branch together some modules in order to get a unit, the inner part of these modules should play no role; actually, the only information required should be the specifications of their interfaces, also called types. The type of a module tells

R. Maieli: Department of Computer Science, University of Rome "La Sapienza";
e-mail: `maieli@dsi.uniroma1.it`.

Q. Puite: Department of Philosophy, University "Roma Tre";
e-mail: `puite@logica.uniroma3.it`

This work was carried out at the University "Roma Tre", in the framework of the European TMR Research Programme "Linear Logic in Computer Science". The authors are grateful to Paul Ruet and to the anonymous referee for their useful comments and remarks on a previous version of this paper.

Correspondence to: R. Maieli: Department of Computer Science, University of Rome "La Sapienza"; e-mail: `maieli@dsi.uniroma1.it`.

Send offprint requests to: Roberto Maieli

Key words or phrases: Linear logic – Proof-nets – Modules – Weak (or linear) distributivity

us about the behavior of this module. In particular, it says that if the branching of modules is done according to their types then the correctness of the whole resulting structure will only depend on the local correctness of the involved modules. At the end of the 80ths J.-Y. Girard has shown in [Gir87a] that proof-nets of linear logic [Gir87] (at least the multiplicative fragment) are suitable "in order to prove that the respect of specification implies correctness". Let us recall the question of modularity of proof-nets, as formulated in [Gir87a]:

> "Assume I am given a program P and I cut it in two parts, *arbitrarily*. I create two (very bad) modules, linked together by their *border*. How can I express that my two modules are complementary, in other terms, that I can branch them by identification of their common border? One would like to define the *type* of the modules as their plugging instructions: these plugging instructions should be such that they authorize the restoring of the original P."

Formally a module of multiplicative linear logic is a structure with a specified border, consisting of all of the hypotheses and some of the conclusions. The conclusions not belonging to the border are called the proper conclusions (see Section 2). Our main question concerns how to code the behavior of a module as a *function* on the border only.

By a slight modification of the original terminology adopted in [DR89] we call *pretype* of a module the set of partitions over the border induced by the systems of Danos-Regnier switchings. Then the type of a module is defined as the double orthogonal of its pretype, according to [Gir87a][1] (see Section 3). The type is an optimal notion describing the behavior of a module: two modules behave in the same way precisely when they have same type.

We now give a characterization of the type of a module which does not make explicit mention to the notion of orthogonality. This characterization only relies on a constructive procedure which allows us to calculate the type of a module, simply by starting from its Danos-Regnier type, here called pretype. The procedure iterates inside the module some elementary steps of graph rewriting, illustrated in Section 4 and only based on the associativity and commutativity of $\otimes$ and $\wp$ and the weak-distributivity laws. The weak-distributive laws correspond to the following (well known) theorems of linear logic: $A \otimes (B \parr C) \vdash (A \otimes B) \parr C$ and $A \otimes (B \parr C) \vdash (A \otimes C) \parr B$ (see [CS97], [AJ94] and Section 8 for a discussion of these works).

In order to show that our rewrite method is complete w.r.t. the type of a module we cut our reasoning in two parts: in Section 6 we study the case when a module is a formula-tree, then in Section 7 we discuss the general case (a module with axioms).

We claim, in Section 8, that our characterization of the type of module can have nice applications to the design of (distributed) theorem provers based on proof-nets.

[1] In [Gir87a] is given an other definition of the type of a module in terms of *permutations* instead of partitions of the border formulas. This choice is justified by the different style of the correctness criterion of proof-nets formulated in terms of *trip* over the switched proof-structures.

Moreover, it implements only some purely geometrical properties on proof-nets, dislike to other methods which rely on the sequent calculus like, for instance, the so called method of "the organization of a formula-tree", due to Danos-Regnier and illustrated in Section 5. For this reason our method could be easily extended to characterize the type of modules of other fragments of linear logic like the non-commutative one [AR00], for which the Danos-Regnier method fails as shown in Section 8.

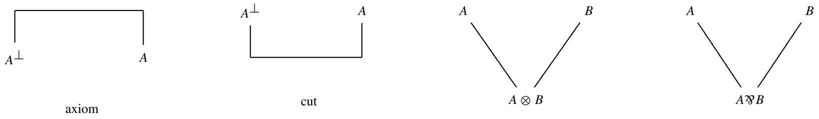
2. The multiplicative fragment of linear logic

The class of formulas of the multiplicative fragment of linear logic (MLL) is defined as the smallest class containing the sets of atoms $\alpha_1, \alpha_2, \ldots$ and their formal negations $\alpha_1^\perp, \alpha_2^\perp, \ldots$, and closed under the multiplicative disjunction $\parr$ (*par*) and conjunction $\otimes$ (*tensor*). Negation $(-)^\perp$ is defined by $(\alpha)^\perp := \alpha^\perp$; $(\alpha^\perp)^\perp := \alpha$, $(A \parr B)^\perp := A^\perp \otimes B^\perp$ and $(A \otimes B)^\perp := A^\perp \parr B^\perp$. Sequents are multisets of formulas, and the inference rules are the axiom-rule, cut-rule, $\otimes$-rule and $\parr$-rule:

$$\frac{}{\vdash A, A^\perp}\ \text{id} \qquad\qquad \frac{\vdash \Gamma, A \qquad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}\ \text{cut}$$

$$\frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ \otimes \qquad\qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \parr$$

A *structure* is a graph consisting of axiom-links, cut-links, $\otimes$-links and $\parr$-links



such that each formula is at most once the conclusion of a link and at most once the premise of a link. A formula which is not conclusion of any other link is called *hypothesis*; whereas a formula which is not premise of any link is called *conclusion*. A *proof-structure* is a structure without hypotheses. To each derivation π we can associate a proof-structures $\overline{\pi}$ with the same conclusions as π, in which case we call π the sequentialization of $\overline{\pi}$. A *module* is a structure with a specified *border*, consisting of all of the hypotheses and some of the conclusions. The conclusions not belonging to the border are called the *proper conclusions*.

A Danos-Regnier *switching* (shortly, DR-switching) of a structure is a choice for one of the two premises of each $\parr$-link, and the corresponding *correction graph* is obtained by erasing the other premise. The *correctness criterion* says that a proof-structures is a *proof-net* iff all correction graphs are trees. Danos and Regnier have also shown that *a proof-structure is a proof-net if and only if it is sequentializable* (Sequentialization Theorem, [DR89]).

3. The Danos-Regnier type of a module

3.1. Generalized partitions

For a module M with set of border formulas B (usually taken $\{1, 2, \dots, n\}$ by a coding), each DR-switching s results in a correction graph with m connected components $\Xi_1, \dots, \Xi_m$ and possibly some cycles. Enumerating for each component Ξ_k the border formulas $i_k^1, \dots, i_k^{n_k}$ it contains and indicating the number c of elementary cycles, we obtain the generalized partition

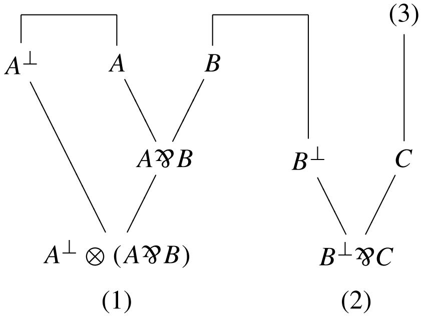
$$p_{M,s} = \left\{\{i_1^1, \dots, i_1^{n_1}\}, \dots, \{i_k^1, \dots, i_k^{n_k}\}, \dots, \{i_m^1, \dots, i_m^{n_m}\}\right\}_c$$

over Binduced by switching s. If a component Ξ_k does not meet the border then $n_k = 0$ whence the corresponding class in $p_{M,s}$ equals $\{\}$. There may be several such classes, hence formally $p_{M,s}$ is a multiset.

Erasing an edge either decreases c by 1 or increases the number of components by 1 (c remaining unchanged), which shows that $\chi_{M,s} := m - c$ is constant for all switchings whence only depending on M. In particular, if all correction graphs are acyclic, $m = |p_{M,s}|$ is invariant under s.

The *pretype* P_M of M is the set of all these induced generalized partitions.

Example 1. The module



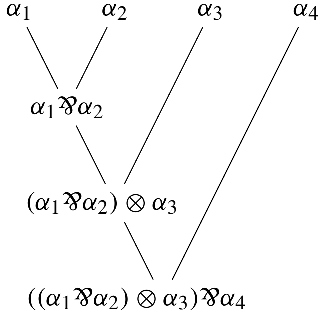
has pretype

$$\left\{ \{\{1\}, \{2, 3\}\}_0\, , \; \{\{1, 2\}, \{3\}\}_0\, , \; \{\{1\}, \{\}, \{2, 3\}\}_1\, , \; \{\{1\}, \{2\}, \{3\}\}_1 \right\}.$$

Every formula $F = F(\alpha_{i_1}, \dots, \alpha_{i_n})$ with n different atomic subformulas corresponds with a module (also written F) with proper conclusion F and border consisting of the hypotheses $\alpha_{i_1}, \dots, \alpha_{i_n}$ (coded by $i_1, \dots, i_n$). Its pretype P_F is a set of ordinary partitions over $\{i_1, \dots, i_n\}$.

The pretype of the formula-tree module $((\alpha_1 \parr \alpha_2) \otimes \alpha_3) \parr \alpha_4$ is

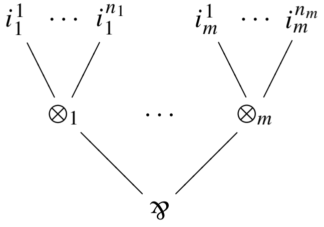
$$\left\{ \{\{1, 3\}, \{2\}, \{4\}\}\, , \; \{\{1\}, \{2, 3\}, \{4\}\} \right\}.$$


α_1
α_2
α_3
α_4
$\alpha_1 \parr \alpha_2$
$(\alpha_1 \parr \alpha_2) \otimes \alpha_3$
$((\alpha_1 \parr \alpha_2) \otimes \alpha_3) \parr \alpha_4$

Given an ordinary partition

$$p = \{\{i_1^1, \ldots, i_1^{n_1}\}, \ldots, \{i_k^1, \ldots, i_k^{n_k}\}, \ldots, \{i_m^1, \ldots, i_m^{n_m}\}\}$$

over $\{1, \ldots, n\}$ ($n = \sum_{k=1}^m n_k$) let us consider the formula $\hat{p} := A_1 \parr (A_2 \parr (\ldots (A_{m-1} \parr A_m) \ldots))$ where each A_k equals $\alpha_{i_k^1} \otimes (\alpha_{i_k^2} \otimes (\ldots (\alpha_{i_k^{n_k-1}} \otimes \alpha_{i_k^{n_k}}) \ldots))$. This formula (considered as module) has a singleton pretype, consisting in the partition p. More general, we call a *bipole* any formula-tree, where no $\parr$-link is in the scope of a $\otimes$-link. In other words, a bipole is a generalized $\parr$-link (eventually with arity 0) of generalized $\otimes$-links, like in the next figure (see also [And01]):



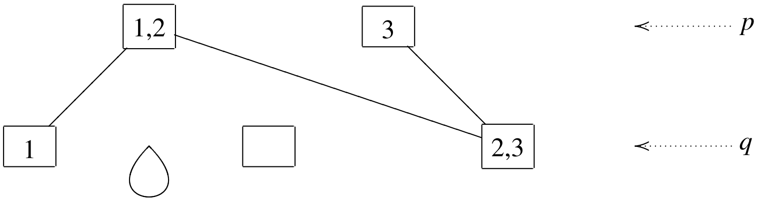
Up to associativity and commutativity of $\otimes$ and $\parr$ the formula $\hat{p}$ is the unique bipole satisfying $P_{\hat{p}} = \{p\}$.

Remark 1. Apart from the allowed emptyness of some classes used to represent some border disconnected components, our definition generalizes the ordinary notion of partition (see Section 2.1 in [DR89]) by the indication of elementary cycles, which enables us to code modules which have some cyclic correction graphs. But, in order to prove the main results of this paper it is enough to consider the ordinary notion of partition; so replacing our definition by the original one should be considered harmless.

3.2. Orthogonality

Given two generalized partitions p and q over $\{1, \ldots, n\}$, let us define $G(p, q)$ as the graph which has for vertices the classes of p and q, and an edge between two classes if they share a point and moreover a circuit for every elementary cycle.

Example 2. $G(\, \big\{\{1,2\},\{3\}\big\}_0\, ,\ \big\{\{1\},\{\},\{2,3\}\big\}_1\,)$ equals



Two partitions p and q are *orthogonal* (notation $p \perp q$) iff $G(p,q)$ is a tree. In this case p and q are acyclic ($c = 0$) and $G(p,q)$ consists of n edges and hence (being a tree) $n+1$ vertices, i.e. $|p| + |q| = n+1$. Moreover, if the border is non-empty ($n > 0$) p (as well as q) contains only non-empty classes ($\forall k : n_k > 0$) and is hence an ordinary partition.

Two sets of generalized partitions P and Q over $\{1,\ldots,n\}$ are said to be orthogonal (notation $P \perp Q$) iff they are pointwise orthogonal. Given a set P of partitions over $\{1,\ldots,n\}$, we write $P^\perp$ for the maximal such Q, i.e. for the set of partitions over $\{1,\ldots,n\}$ that are orthogonal to all the elements of P. (For empty P we should specify n, notation $P = \emptyset_n$, in which case $P^\perp = \emptyset_n^\perp$ is nothing else than the set of all generalized partitions over $\{1,\ldots,n\}$. This set is infinite, since it contains e.g. all $\big\{\{1,2,\ldots,n\}\big\}_c$ ($c \in \mathbb{N}$).)

We call P a *fact* iff $P = P^{\perp\perp}$.

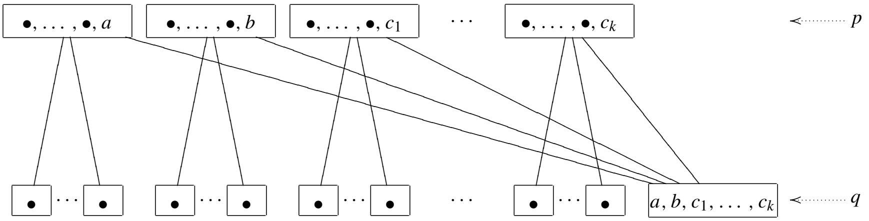
Lemma 3. *Let A, B and the A_i be sets of generalized partitions over $\{1,\ldots,n\}$. Then the following hold:*

1. $A \perp \emptyset_n$;
2. $A \perp B$ *iff* $A \subseteq B^\perp$ *iff* $B \subseteq A^\perp$;
3. $A \subseteq B$ *implies* $B^\perp \subseteq A^\perp$;
4. $A \subseteq A^{\perp\perp}$;
5. $A \perp B$ *iff* $A^{\perp\perp} \perp B^{\perp\perp}$;
6. $A^\perp = A^{\perp\perp\perp}$;
7. ($A \perp B$ *and* $A^\perp \perp B^\perp$) *iff* ($A^\perp = B^{\perp\perp}$) *iff* ($B^\perp = A^{\perp\perp}$);
8. *In case* $A \perp B$ *and* $A^\perp \perp B^\perp$: ($B^\perp = A$ *and* $B = A^\perp$) *iff* ($A^{\perp\perp} = A$ *and* $B = B^{\perp\perp}$);
9. A *is a fact iff* $\exists B : A = B^\perp$;
10. $(\bigcup_i A_i)^\perp = \bigcap_i A_i^\perp$;
11. $(\bigcap_i A_i)^\perp \supseteq \bigcup_i A_i^\perp$.

From Lemma 3 we infer that an arbitrary intersection of facts is a fact. For a union this does not hold; otherwise each P would be a fact as $P = \bigcup_{p\in P}\{p\}$, while all singleton pretypes are facts, by the following Lemma 4. Every A is contained in the fact $A^{\perp\perp}$. Actually, this is the smallest fact A is contained in: if $A \subseteq B^\perp$ then $A^{\perp\perp} \subseteq B^\perp$.

Lemma 4. *The pretype of a bipole $\hat{p}$ is a fact: $P_{\hat{p}} = \{p\} = P_{\hat{p}}^{\perp\perp}$.*

Proof. $P_{\hat{p}} \subseteq P_{\hat{p}}^{\perp\perp}$ trivially holds (Lemma 3). For the other way around, suppose $p' \in P_{\hat{p}}^{\perp\perp}$; we have to prove that $p' \in P_{\hat{p}} = \{p\}$, i.e. that $p' = p$. Given $a, b \in \{1, \dots, n\}$ which do not belong to the same class in p there is a $q \in P_{\hat{p}}^{\perp}$ for which they do belong to the same class, e.g. a partition consisting of $n - |p|$ singletons and one class of cardinality $|p|$ containing one element of each class of p (see figure below). As $\{p'\} \perp P_{\hat{p}}^{\perp}$, in particular $p' \perp q$, whence also in p' the elements a and b must belong to distinct classes to avoid a cycle. So p' is a refinement of p, obtained by splitting some classes of p. But $|p| = n + 1 - |q| = |p'|$ so p' is the trivial refinement of p, viz. p.



□

From the definition of proof-net (all correction graphs are trees) we directly infer that two modules connected along their common border constitute a proof-net iff their respective pretypes are orthogonal:

Proposition 5. *Let M and M' be two modules with border $\{1, \dots, n\}$ resp. $\{1', \dots, n'\}$. Then $M \amalg_{i \sim i'} M'$ is a proof-net iff $P_M \perp P_{M'}$.*

Reformulating the right hand side as $P_M^{\perp\perp} \perp P_{M'}^{\perp\perp}$, we see that the interaction of M with other modules M' is completely determined by $P_M^{\perp\perp}$, whence we define the *type* T_M of M as this bi-orthogonal:

$$T_M := P_M^{\perp\perp}.$$

Two modules have the same type iff they behave the same:

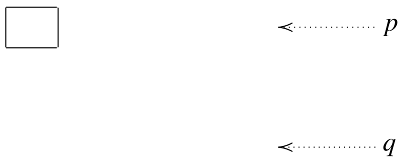
Proposition 6. *Let M_1 and M_2 be two modules with border $\{1, \dots, n\}$, $n > 0$. Then $T_{M_1} = T_{M_2}$ if and only if for all modules M' (with border $\{1', \dots, n'\}$) $M_1 \amalg_{i \sim i'} M'$ is a proof-net precisely if $M_2 \amalg_{i \sim i'} M'$ is a proof-net.*

Proof. The (only if)-part is immediate: suppose $P_{M_1}^{\perp\perp} = P_{M_2}^{\perp\perp}$, then $P_{M_1}^{\perp\perp} \perp P_{M'}^{\perp\perp}$ iff $P_{M_2}^{\perp\perp} \perp P_{M'}^{\perp\perp}$. The other way around, suppose for all modules M' (with border $\{1', \dots, n'\}$) $M_1 \amalg_{i \sim i'} M'$ is a proof-net precisely if $M_2 \amalg_{i \sim i'} M'$ is a proof-net. We will prove that $P_{M_1}^{\perp} \subseteq P_{M_2}^{\perp}$, implying (by symmetry) that $P_{M_1}^{\perp} = P_{M_2}^{\perp}$ and hence that $P_{M_1}^{\perp\perp} = P_{M_2}^{\perp\perp}$. Given $p \in P_{M_1}^{\perp}$, by non-emptyness of P_{M_1} there is a $q \in P_{M_1}$ such that $p \perp q$. So p is an ordinary partition, and let $\hat{p}$ be the bipole defined in

Example 1, to which we attach some axiom-links turning hypothesis border into conclusion border if necessary. As $P_{\hat{p}} = \{p\} \subseteq P^{\perp}_{M_1}$ we know $P_{\hat{p}} \perp P_{M_1}$ whence by assumption also $P_{\hat{p}} \perp P_{M_2}$, implying $p \in P^{\perp}_{M_2}$. □

3.3. Empty border

Generalized partitions provide a nice interpretation for the case with empty border ($n = 0$). In general, a module M without border will have pretype consisting of generalized partitions of the form $\big\{\{\}, \dots, \{\}\big\}_c$ depending on the number of connected components m in a correction graph and the number of elementary cycles c. A module M without border is a proof-net iff $P_M = \left\{ \big\{\{\}\big\}_0 \right\}$, i.e. iff every correction graph is a tree (one component, no cycles). The empty module $\varnothing$ has only one correction graph (zero components), whence $P_{\varnothing} = \left\{ \{\}_0 \right\}$. As $G(\ \big\{\{\}\big\}_0\ ,\ \{\}_0\ )$ equals



we have $\big\{\{\}\big\}_0 \perp \{\}_0$ whence

$$\left\{ \big\{\{\}\big\}_0 \right\} \perp \left\{ \{\}_0 \right\} \tag{1}$$

i.e. $P_{\Pi} \perp P_{\varnothing}$ for Π a proof-net and $\varnothing$ the empty module. By Proposition 5 this means that $\Pi \amalg \varnothing$ is a proof-net. Observe that (1) has nothing to do with $P \perp \emptyset_0$ which holds trivially (for any set P of generalized partitions over the empty border) as the universal quantifier ranges over the empty set (of partitions over the empty border).

3.4. Correctness

We call a module M *correct* iff $P^{\perp}_{M} \neq \emptyset_n$, implying (in case $n > 0$) that P_M consists only in ordinary (not generalized) partitions. E.g., the module in the Example 1 is not correct. A non-empty module with empty border is correct iff it is a proof-net.

Every substructure M of a proof-net Π is a correct module, even when $M = \Pi$ and the border is empty (by (1)). The other way around every correct module is always completable to a proof-net (e.g. in case $n > 0$ by attaching $\hat{p}$ where $p \in P^{\perp}_{M}$). A module is incorrect iff $P^{\perp}_{M} = \emptyset_n$, i.e. iff $T_M = P^{\perp\perp}_{M} = \emptyset^{\perp}_{n}$. So for fixed n all incorrect modules have the same type, implying by Proposition 6 that they behave the same; indeed, none of them forms a proof-net with any module M'.

3.5. The pretype of a formula-tree

In Example 1 we defined the pretype of a formula-tree, considered as module: every formula $F = F(\alpha_{i_1}, \dots, \alpha_{i_n})$ with n different atomic subformulas corresponds with a module (also written F) with proper conclusion F and border consisting of the hypotheses $\alpha_{i_1}, \dots, \alpha_{i_n}$ (coded by $i_1, \dots, i_n$). Its pretype P_F is a set of ordinary partitions over $\{i_1, \dots, i_n\}$.

Theorem 7. *Let $F = F(\alpha_{i_1}, \dots, \alpha_{i_n})$ be a formula-tree with n different atomic subformulas. Then*

$$(P_{F^\perp})^\perp = (P_F)^{\perp\perp}.$$

Proof. According to Lemma 3 it is equivalent to prove

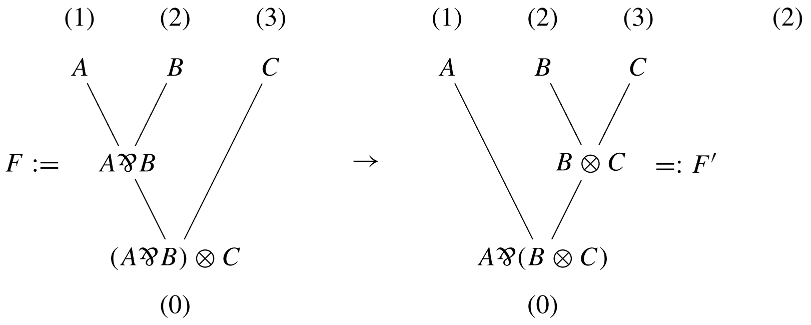
$$P_{F^\perp} \perp P_F \text{ and } P_{F^\perp}^\perp \perp P_F^\perp.$$

The first statement follows from the fact that $F^\perp \amalg F$ is a proof-net (namely the eta-expanded identity). The second relies upon cut elimination: if $p \in P_{F^\perp}^\perp$ and $q \in P_F^\perp$ they are ordinary, and $\hat{p} \amalg P_{F^\perp}$ is a proof-net as well as $\hat{q} \amalg P_F^\perp$. Connecting them by a cut link yields a proof-net with normal form $\hat{p} \amalg \hat{q}$ showing $p \perp q$. □

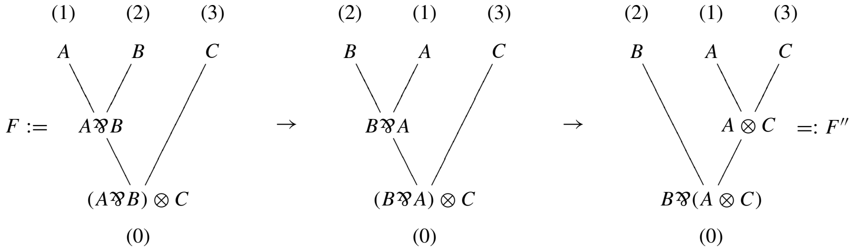
4. A rewrite relation

4.1. Definition

We define a *rewrite relation* on (untyped) structures generated by the associativity and commutativity of $\otimes$ and $\wp$ and the *weak-distributivity*:



In the sequel we assume F and F' like above and F'', obtained by rewriting F in the following way:



Proposition 8. *Let Π be a proof-structures containing a sub-structure F and Π' and Π'' the two proof-structures obtained by replacing F by F', respectively F''. Then Π is a proof-net if and only if Π' and Π'' are proof-nets.*

Proof. **Only-part** - Let Π be a proof-net containing F. By deleting F from Π we obtain a module M with border $\{0, 1, 2, 3\}$, and as $\Pi = M \amalg F$ is a proof-net we know

$$P_M \perp P_F = \Big\{ \big\{\{1\}, \{2, 3, 0\}\big\} \,,\ \big\{\{2\}, \{1, 3, 0\}\big\} \Big\}.$$

So any $q \in P_M$ must have 3 classes, and the elements 0 (inhabiting a class with 2, 3 resp. 1, 3, i.e. with all elements) and 3 must be in singleton classes, whence

$$P_M = \Big\{ \big\{\{1, 2\}, \{3\}, \{0\}\big\} \Big\}.$$

But then also

$$P_M \perp P_{F'} = \Big\{ \big\{\{1\}, \{2, 3, 0\}\big\} \,,\ \big\{\{1, 0\}, \{2, 3\}\big\} \Big\},$$

showing that $\Pi' = M \amalg F'$ is a proof-net.

By analogous reasoning we conclude $\Pi'' = M \amalg F''$ is a proof-net.

If-part - We show F' and F'' together can take over the role of F in the following sense:

$P_{F'} = \Big\{ \big\{\{1\}, \{2, 3, 0\}\big\} \,,\ \big\{\{1, 0\}, \{2, 3\}\big\} \Big\}$ so

$P_{F'}^{\perp} = \Big\{ \big\{\{1, 2\}, \{3\}, \{0\}\big\} \,,\ \big\{\{1, 3\}, \{2\}, \{0\}\big\} \Big\}$, and

$P_{F''} = \Big\{ \big\{\{2\}, \{1, 3, 0\}\big\} \,,\ \big\{\{2, 0\}, \{1, 3\}\big\} \Big\}$ so

$P_{F''}^{\perp} = \Big\{ \big\{\{1, 2\}, \{3\}, \{0\}\big\} \,,\ \big\{\{2, 3\}, \{1\}, \{0\}\big\} \Big\}$, hence

$(P_{F'} \cup P_{F''})^{\perp} = P_{F'}^{\perp} \cap P_{F''}^{\perp} = \Big\{ \big\{\{1, 2\}, \{3\}, \{0\}\big\} \Big\} = P_F^{\perp}.$

So, suppose $\Pi' = M \amalg F'$ and $\Pi'' = M \amalg F''$ are proof-nets, then $P_M \perp P_{F'} \cup P_{F''}$, so $P_M \subseteq (P_{F'} \cup P_{F''})^{\perp} = P_F^{\perp}$ which shows that $\Pi = M \amalg F$ is a proof-net. □

Proposition 9. *Let M be a module and M' and M'' the two modules obtained by replacing a substructure F of M by F', respectively F''. Then $P_M \subseteq P_{M'} \cup P_{M''}$.*

Proof. Observe that all geometrical connections induced by the switchings of M are present in the right-hand side of the inclusion. □

We will now generalize $P_F^\perp = P_{F'}^\perp \cap P_{F''}^\perp = (P_{F'} \cup P_{F''})^\perp$ to arbitrary modules M in which we replace F.

Proposition 10. *Let M be a module and M' and M'' the two modules obtained by replacing a substructure F of M by F' respectively F''. Then $P_M^\perp = P_{M'}^\perp \cap P_{M''}^\perp = (P_{M'} \cup P_{M''})^\perp$.*

Proof. Let M be a module containing F and let M' (resp. M'') be the module obtained after replacing F by F' (resp. F''). First we will show that $P_M^\perp \subseteq P_{M'}^\perp$. Given $p \in P_M^\perp$, then $\{p\} \perp P_M$ and p is an ordinary partition, so $\hat{p} \amalg M$ is a proof-net. By Proposition 8 also $\hat{p} \amalg M'$ is a proof-net, implying $\{p\} \perp P_{M'}$ whence $p \in P_{M'}^\perp$. Similarly $P_M^\perp \subseteq P_{M''}^\perp$, whence $P_M^\perp \subseteq P_{M'}^\perp \cap P_{M''}^\perp$.

The other way around, suppose $p \in P_{M'}^\perp \cap P_{M''}^\perp$, then both $\hat{p} \amalg M'$ and $\hat{p} \amalg M''$ are proof-nets, whence also $\hat{p} \amalg M$ is a proof-net, implying $p \in P_M^\perp$. □

The property that $P_M^\perp \subseteq P_{M'}^\perp$ also implies that $P_{M'} \subseteq P_{M'}^{\perp\perp} \subseteq P_M^{\perp\perp} = T_M$, so the pretypes of the reducts of M remain in the type of M, i.e.

Lemma 11. *Let $\rightarrow$ be the rewrite relation defined by 2, associativity and commutativity of $\otimes$ and $\invamp$. Let M be a module. Then*

$$\bigcup_{N:M\rightarrow^* N} P_N \subseteq T_M.$$

Remark 2. Our aim is to show the inclusion of Lemma 11 to be an equality, i.e.

$$P_M^{\perp\perp} \subseteq \bigcup_{N:M\rightarrow^* N} P_N.$$

We know $P_M^\perp = P_{M'}^\perp \cap P_{M''}^\perp$, and if from this we could conclude $P_M^{\perp\perp} = P_{M'}^{\perp\perp} \cup P_{M''}^{\perp\perp}$ we would be done since then

$$P_M^{\perp\perp} = P_{M'}^{\perp\perp} \cup P_{M''}^{\perp\perp} = \bigcup_{N:M'\rightarrow^* N} P_N \cup \bigcup_{N:M''\rightarrow^* N} P_N \subseteq \bigcup_{N:M\rightarrow^* N} P_N$$

by induction hypothesis (assuming we may reason by induction). However, from $P_M^\perp = P_{M'}^\perp \cap P_{M''}^\perp$ we may only conclude $P_M^{\perp\perp} \supseteq P_{M'}^{\perp\perp} \cup P_{M''}^{\perp\perp}$ and this inclusion may be strict. This in turn means that the inclusion

$$\bigcup_{N:M'\rightarrow^* N} P_N \cup \bigcup_{N:M''\rightarrow^* N} P_N \subseteq \bigcup_{N:M\rightarrow^* N} P_N$$

may be strict.

Consider the following example: let M be the formula $((1 \otimes 2)\parr 3) \otimes (4 \parr 5)$ and $M' = (((1 \otimes 2)\parr 3) \otimes 4)\parr 5$, $M'' = (((1 \otimes 2)\parr 3) \otimes 5)\parr 4$. M' and M'' are the reducts obtained from M after we apply one step of 2, associativity and commutativity. Then

$$\begin{aligned}\bigcup_{N:M\to^* N} P_N =&\{3\parr 4\parr (1 \otimes 2 \otimes 5), 3\parr 5\parr (1 \otimes 2 \otimes 4), 3\parr (1 \otimes 4)\parr (2 \otimes 5),\\ &3\parr (1 \otimes 5)\parr (2 \otimes 4), 4\parr (1 \otimes 2)\parr (3 \otimes 5), 5\parr (1 \otimes 2)\parr (3 \otimes 4)\};\end{aligned}$$

$$\bigcup_{N:M'\to^* N} P_N =\{3\parr 5\parr (1 \otimes 2 \otimes 4), 5\parr (1 \otimes 2)\parr (3 \otimes 4)\};$$

$$\bigcup_{N:M''\to^* N} P_N =\{3\parr 4\parr (1 \otimes 2 \otimes 5), 4\parr (1 \otimes 2)\parr (3 \otimes 5)\}.$$

Clearly the union of latter two sets is strictly included in the former set. We can interpret this fact as follows: it is not true that every rewrite sequence may be converted into one starting (up to associativity and commutativity) with $M \to M'$ or $M \to M''$; possibly we first have to apply the 2 to another $\parr$-link.

4.2. A reducing complexity for formula-trees

Suppose M is a formula-tree. We will define the complexity of M in such a way that it strictly decreases under the 2 and remains constant under associativity and commutativity of $\otimes$ and $\parr$.

For a formula-tree M and a node v we define p_v as the number of $\parr$-links in the subtree with root v and we define t_v similarly as the number of $\otimes$-links in the subtree with root v. Denoting the root of M also by M, with this notation p_M (resp. t_M) equals the total number of $\parr$ (resp. $\otimes$) links occurring in M.

We define the *complexity* $c(M)$ of M by

$$c(M) = \sum_{l \text{ a } \otimes\text{-link of } M} p_{l_1}(1 + t_{l_2}) + p_{l_2}(1 + t_{l_1}),$$

where l_1 and l_2 are the two premises of l.

If $c(M) = 0$ either the sum is empty (i.e. there are no $\otimes$-links) or for every $\otimes$-link l the summand $p_{l_1}(1 + t_{l_2}) + p_{l_2}(1 + t_{l_1}) = 0$, i.e. both p_{l_1} and p_{l_2} vanish. This means no $\parr$'s dominate a $\otimes$, so M is a bipole as defined in Example 1: a generalized $\parr$ of generalized $\otimes$.

Lemma 12. *$c(M) \leq p_M t_M$, and $c(M) = p_M t_M$ iff M is the negation of a bipole: a generalized $\otimes$ of generalized $\parr$.*

Proof. By induction on M. For M an atom the result holds.

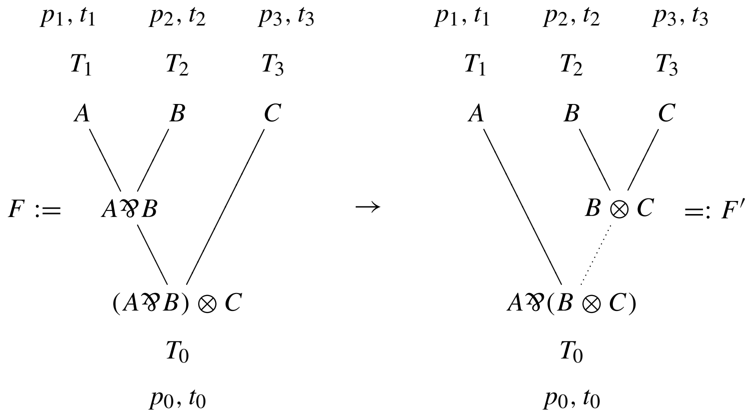
If $M = M_1 \parr M_2$ then $c(M) = c(M_1) + c(M_2) \leq p_{M_1}t_{M_1} + p_{M_2}t_{M_2}$ while $p_M t_M = (p_{M_1} + 1 + p_{M_2})(t_{M_1} + t_{M_2}) = p_{M_1}t_{M_1} + p_{M_2}t_{M_2} + (1 + p_{M_2})t_{M_1} +$

$(p_{M_1}+1)t_{M_2} \geq p_{M_1}t_{M_1}+p_{M_2}t_{M_2}$. Equality holds iff it holds for M_1 and M_2 (so iff they are negated bipoles) and iff M_1 and M_2 are $\otimes$-free, i.e. iff M is a generalized $\parr$.

If $M = M_1 \otimes M_2$ then $c(M) = c(M_1)+c(M_2)+p_{M_1}(1+t_{M_2})+p_{M_2}(1+t_{M_1}) \leq p_{M_1}t_{M_1} + p_{M_2}t_{M_2} + p_{M_1}(1+t_{M_2}) + p_{M_2}(1+t_{M_1}) = p_M t_M$. Equality holds iff it holds for M_1 and M_2 (so iff each of them is a generalized $\otimes$ of generalized $\parr$), i.e. iff M is. □

Lemma 13. *Let M be a formula-tree and M' the formula-tree obtained by replacing a substructure F by F'. Then* $0 \leq c(M') < c(M)$.

Proof. The substructure F defines 4 sub-trees T_i of M, and we denote the number of $\parr$ and $\otimes$ in T_i by p_i and t_i.

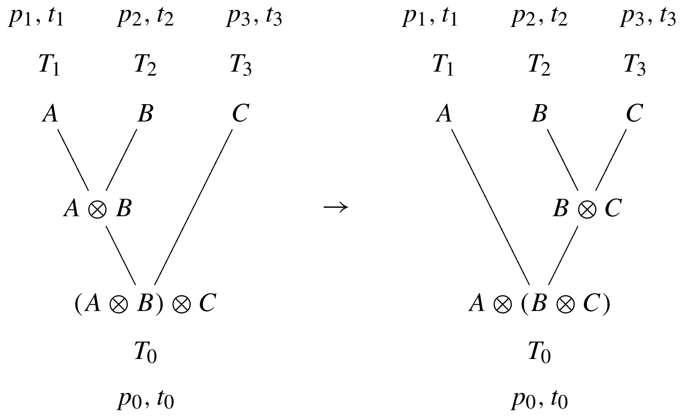


In the summand of $c(M) = \sum_{l \text{ a } \otimes\text{-link of } M} p_{l_1}(1+t_{l_2}) + p_{l_2}(1+t_{l_1})$ we consider the different cases for l. If l is the $\otimes$-link belonging to F, the summand $(1+p_1+p_2)(1+t_3)+p_3(1+t_1+t_2)$ becomes $p_2(1+t_3)+p_3(1+t_2)$ for $c(M')$ whence reduces by $(1+p_1)(1+t_3)+p_3t_1 \geq 1$. For all other $\otimes$-links l the summand is invariant; this is clear when l belongs to T_1, T_2 or T_3, but also for l belonging to T_0 the involved numbers do not changes (although the involved subtrees may change by the rewriting). □

The proof shows that $c(M)$ reduces by at least one. It reduces by 1 if $p_1 = 0 = t_3$ and either $p_3 = 0$ or $t_1 = 0$.

Lemma 14. *Let M be a formula-tree and M' the formula-tree obtained by applying associativity or commutativity for $\otimes$ or $\parr$. Then* $0 \leq c(M') = c(M)$.

Proof. We consider first associativity for $\otimes$. The 2 $\otimes$-links define 4 sub-trees T_i of M, and we denote the number of $\parr$ and $\otimes$ in T_i by p_i and t_i.



The two indicated $\otimes$-links in the summation $c(M) = \sum_{l \text{ a } \otimes\text{-link of } M} p_{l_1}(1+t_{l_2}) + p_{l_2}(1+t_{l_1})$ lead to $p_1(1+t_2)+p_2(1+t_1)+(p_1+p_2)(1+t_3)+p_3(1+t_1+1+t_2) = 2(p_1+p_2+p_3)+p_1(t_2+t_3)+p_2(t_1+t_3)+p_3(t_1+t_2)$ and they also lead to this value in the summation $c(M')$. For all other $\otimes$-links l the summand is invariant; this is clear when l belongs to T_1, T_2 or T_3, but also for l belonging to T_0 the involved numbers do not changes (although the involved subtrees may change by the rewriting).

For commutativity for $\otimes$ observe that $p_{l_1}(1+t_{l_2})+p_{l_2}(1+t_{l_1})$ is symmetric. For associativity or commutativity for $\parr$ all summands in $c(M)$ are clearly invariant. □

4.3. Alternative definition of complexity

Given a vertex v of M, we call its complexity $c(v)$ the number of $\otimes$-links strictly below, and all $\otimes$-links above the other premise of such a $\otimes$-link below. Alternatively said:

$$c(v) := \#\{t \text{ a } \otimes\text{-link of } M \mid t \wedge v \text{ is a } \otimes\text{-link strictly below } v\}.$$

The complexity of a link is the complexity of its conclusion vertex. Observe that $c(v)$ is the number of $\otimes$-links not above or equal to v, which are reachable from v by an appropriate DR-switching.

We define

$$c'(M) = \sum_{p \text{ a } \parr\text{-link of } M} c(p).$$

Proposition 15. $c(M) = c'(M)$.

Proof.

$$
\begin{aligned}
c'(M) &= \sum_{p \text{ a } \parr\text{-link of } M} \#\{t \text{ a } \otimes\text{-link of } M \mid t \wedge p \text{ is a } \otimes\text{-link below } p\} \\
&= \sum_{p \text{ a } \parr\text{-link of } M} \sum_{t \text{ a } \otimes\text{-link of } M} \chi(t \wedge p \text{ is a } \otimes\text{-link below } p) \\
&= \sum_{l \text{ a } \otimes\text{-link of } M} \sum_{p \succ l \text{ a } \parr\text{-link of } M} \sum_{t \text{ a } \otimes\text{-link of } M} \chi(l = t \wedge p) \\
&= \sum_{l \text{ a } \otimes\text{-link of } M} \left(\left(\sum_{p \succeq l_1 \text{ a } \parr\text{-link of } M} \sum_{(t = l) \vee (t \succeq l_2) \text{ a } \otimes\text{-link of } M} 1 \right) + \right. \\
&\quad + \left. \left(\sum_{p \succeq l_2 \text{ a } \parr\text{-link of } M} \sum_{(t = l) \vee (t \succeq l_1) \text{ a } \otimes\text{-link of } M} 1 \right) \right) \\
&= \sum_{l \text{ a } \otimes\text{-link of } M} p_{l_1}(1 + t_{l_2}) + p_{l_2}(1 + t_{l_1}) = c(M).
\end{aligned}
$$

□

For some of the proofs in the previous subsection this definition shortens the reasoning considerably. E.g.

$$c'(M) = \sum_{p \text{ a } \parr\text{-link of } M} c(p) \leq \sum_{p \text{ a } \parr\text{-link of } M} t_M \leq p_M t_M$$

yields Lemma 12 again.

For Lemma 13, we distinguish the different ⅋-links in the summation.
- If p is the ⅋-link occurring in F, $c(p) = t'_0 + 1 + t_3$ which reduces to t'_0.
- If p is one of the p_1 ⅋-links occurring in T_1, $c(p) = t'_1 + t'_0 + 1 + t_3$ which reduces to $t'_1 + t'_0$.
- If p is one of the p_2 ⅋-links occurring in T_2, $c(p) = t'_2 + t'_0 + 1 + t_3$ which remains $t'_2 + t'_0 + 1 + t_3$.
- If p is one of the p_3 ⅋-links occurring in T_3, $c(p) = t'_3 + t'_0 + 1 + t_1 + t_2$ which reduces to $t'_3 + t'_0 + 1 + t_2$.
- If p is one of the p_0 ⅋-links occurring in T_0, $c(p)$ is unaltered, although an involved subtree may alter.

In total, $c'(M)$ hence reduces by $1 + t_3 + p_1(1 + t_3) + p_3 t_1$ as we saw before.

It is immediate that the complexity of a ⅋-link p only depends on the ⅋-cluster it belongs to; all other ⅋'s in this cluster have the same complexity. Moreover, the complexity of a ⅋-link does not change if we modify certain ⊗-clusters. Hence we immediately obtain Lemma 14.

5. The organization of a formula-tree module

In this section we consider MLL′, multiplicative linear logic with arbitrary (but disjoint) atomic axioms. The rules are given by

$$\frac{}{\vdash \Xi}\ \text{ax} \qquad \frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ \otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \parr$$

where Ξ stands for one or more different positive atoms α_k. We require that all atoms occurring in a sequent are different, yielding a restriction on the applicability of the two logical rules.

The notion of proof-structures is defined as before (we allow the general axiom links and no cut links). A proof-structures is a proof-net iff the Danos-Regnier correctness criterion holds iff its sequentializable in MLL$'$.

If a derivation π derives a formula $F = F(\alpha_{i_1}, \dots, \alpha_{i_n})$, then the axioms of π induce a proper partition $o_{F,\pi}$ over $\{i_1, \dots, i_n\}$: two indices are in the same class iff the corresponding atoms are a conclusion of the same axiom. The *organization* O_F of F is the set of all these organizations $o_{F,\pi}$. E.g.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 1, 3, 4}}{\vdash 1, 3\parr 4} \qquad \dfrac{}{\vdash 2}}{\vdash 1 \otimes 2, 3\parr 4}}{\vdash (1 \otimes 2)\parr(3\parr 4)} \quad \text{has organization } \big\{\{1,3,4\},\{2\}\big\}$$

and considering all other derivations of $F = (1 \otimes 2)\parr(3\parr 4)$ we find that

$$O_F = \Big\{\ \big\{\{1,3,4\},\{2\}\big\}\ ,\ \big\{\{1,3\},\{2,4\}\big\}\ ,\ \big\{\{1,4\},\{2,3\}\big\}\ ,\ \big\{\{1\},\{2,3,4\}\big\}\ \Big\}.$$

Lemma 16. *Given an ordinary partition p then:*

$$P_{\hat{p}} = \{p\} = O_{\hat{p}^\perp}.$$

Proof. The left equation follows from the definition of bipole (see Example 1). For the right equation observe that p corresponds to the organization of the derivation of the formula $\hat{p}^\perp$. □

We will now show the relation between $O_{F^\perp}$ and $O_{F'^\perp}$ for reducts F' of F under the rewrite relation of the previous section: 2-rule, associativity and commutativity of $\otimes$ and $\parr$. In particular it follows from the first lemma that if $F \to^* \hat{p}$ we obtain a particular element of $O_{F^\perp}$: $O_{\hat{p}^\perp} = \{p\} \subseteq O_{F^\perp}$.

Lemma 17. *If $F \to F'$ then $O_{F'^\perp} \subseteq O_{F^\perp}$.*

Proof. Let $p \in O_{F'^\perp}$ be given, say $p \in o_{F'^\perp, \pi'}$.
In case the rewrite step is the 2 then π' is of the form as indicated below on the left. We can transform it to a proof π of $F^\perp = G[(A \otimes B)\parr C]$ with same organization, showing $p \in O_{F^\perp}$:

$$\dfrac{\dots \qquad \dfrac{\vdash \Gamma, A \qquad \dfrac{\begin{matrix} \dots \qquad \dfrac{\vdash \Sigma, B, C}{\vdash \Sigma, B\parr C}\ \parr \\ \vdots \qquad\qquad \vdots \end{matrix}}{\Delta, B\parr C}}{\vdash \Delta, \Gamma, A \otimes (B\parr C)}\ \otimes \\ \vdots \qquad\qquad \vdots}{\vdash G[A \otimes (B\parr C)]} \qquad \text{becomes} \qquad \dfrac{\dots \qquad \dfrac{\begin{matrix} \dots \qquad \dfrac{\dfrac{\vdash \Gamma, A \qquad \vdash \Sigma, B, C}{\vdash \Sigma, \mathbf{\Gamma}, \mathbf{A} \otimes B, C}\ \otimes}{\vdash \Sigma, \mathbf{\Gamma}, (\mathbf{A} \otimes B)\parr C}\ \parr \\ \vdots \qquad\qquad \vdots \end{matrix}}{\vdash \Delta, \mathbf{\Gamma}, (\mathbf{A} \otimes B)\parr C} \\ \vdots \qquad\qquad \vdots}{\vdash G[(A \otimes B)\parr C]}$$

In case the rewrite step is a direction of associativity for $\otimes$, then replacing

$$\begin{array}{c}
\qquad\qquad \dfrac{\vdash \Sigma, B, C}{\cdots \qquad \vdash \Sigma, B\parr C}\parr \\
\dfrac{\vdots \qquad\qquad \vdots}{\dfrac{\vdash \Gamma, A, B\parr C}{\cdots \qquad \vdash \Gamma, A\parr(B\parr C)}\parr} \\
\dfrac{\vdots \qquad\qquad \vdots}{\vdash G[A\parr(B\parr C)]}
\end{array}
\quad \text{by} \quad
\begin{array}{c}
\cdots \qquad \vdash \Sigma, B, C \\
\dfrac{\vdots \qquad\qquad \vdots}{\dfrac{\dfrac{\vdash \Gamma, A, B, C}{\vdash \Gamma, A\parr B, C)}\parr}{\cdots \qquad \vdash \Gamma, (A\parr B)\parr C)}\parr} \\
\dfrac{\vdots \qquad\qquad \vdots}{\vdash G[(A\parr B)\parr C]}
\end{array}$$

yields a proof π of $F^{\perp} = G[(A\parr B)\parr C]$ with same organization, showing $p \in O_{F^{\perp}}$.
In case the rewrite step is a direction of associativity for $\parr$, then replacing

$$\begin{array}{c}
\qquad\qquad \dfrac{\vdash \Sigma, B \qquad \vdash \Pi, C}{\cdots \qquad \vdash \Sigma, \Pi, B\otimes C}\otimes \\
\dfrac{\vdash \Gamma, A \qquad \dfrac{\vdots \qquad\qquad \vdots}{\Delta, B\otimes C}}{\cdots \qquad \vdash \Delta, \Gamma, A\otimes(B\otimes C)}\otimes \\
\dfrac{\vdots \qquad\qquad \vdots}{\vdash G[A\otimes(B\otimes C)]}
\end{array}
\quad \text{by} \quad
\begin{array}{c}
\dfrac{\dfrac{\vdash \Gamma, A \qquad \vdash \Sigma, B}{\vdash \Sigma, \mathbf{\Gamma}, \boldsymbol{A\otimes} B}\otimes \qquad \vdash \Pi, C}{\cdots \qquad \vdash \Sigma, \Pi, \mathbf{\Gamma}, (\boldsymbol{A\otimes} B)\otimes C}\otimes \\
\dfrac{\vdots \qquad\qquad \vdots}{\cdots \qquad \vdash \Delta, \mathbf{\Gamma}, (\boldsymbol{A\otimes} B)\otimes C} \\
\dfrac{\vdots \qquad\qquad \vdots}{\vdash G[(A\otimes B)\otimes C]}
\end{array}$$

yields a proof π of $F^{\perp} = G[(A \otimes B) \otimes C]$ with same organization, showing $p \in O_{F^{\perp}}$.
Finally, in case the rewrite step is commutativity for $\otimes$ or $\parr$, then the result is immediate. □

Lemma 18. *If $F = G \otimes H$ and $c(F) > 0$ then*

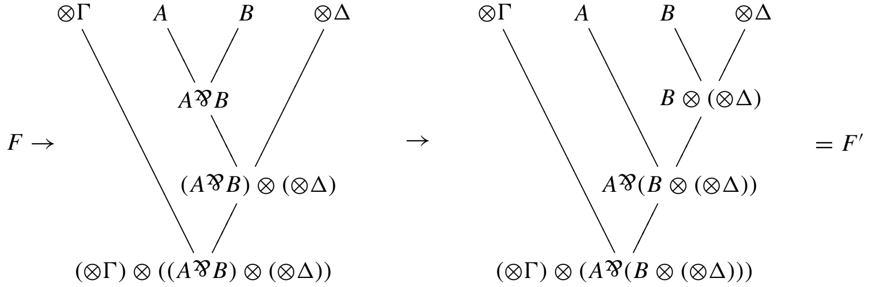
$$O_{F^{\perp}} \subseteq \bigcup_{\substack{F':F\to^{*}F' \\ c(F)>c(F')}} O_{F'^{\perp}}.$$

Proof. Suppose $c(F) > 0$ and let $p \in O_{F^{\perp}}$ be given. We have to show that there is an F' such that $F \to^{*} F'$, $c(F) > c(F')$ and $p \in O_{F'^{\perp}}$.

As $F = G \otimes H$ we know $p = o_{G^{\perp}\parr H^{\perp},\pi}$ for a certain π which we can take negatively focalized:

$$\begin{array}{c}
\vdash \Gamma^{\perp}, A^{\perp} \qquad \vdash B^{\perp}, \Delta^{\perp} \\
\hline
\vdash \Gamma^{\perp}, A^{\perp}\otimes B^{\perp}, \Delta^{\perp} \\
\hline\hline
\vdash G^{\perp}, H^{\perp} \\
\hline
\vdash G^{\perp}\parr H^{\perp}
\end{array}
\begin{array}{l}
\\ \otimes \\ \\ \parr \\ \\ \parr
\end{array}$$

So all formulas in $\vdash \Gamma^{\perp}, A^{\perp} \otimes B^{\perp}, \Delta^{\perp}$ are final-$\parr$ free, i.e. in Γ and Δ there are only atoms or $\parr$'s. Moreover, at least one of the two contexts Γ and Δ is non-empty, say Δ. We choose F' as follows:



where the first step consists of associativity and commutativity. F' is clearly a reduct of F and of smaller complexity by Lemma 13. We can derive $F'^{\perp}$ still by the organization p.

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma^{\perp}, A^{\perp}}{\vdash \parr(\Gamma^{\perp}), A^{\perp}}\,\parr \qquad \dfrac{\dfrac{\vdash B^{\perp}, \Delta^{\perp}}{\vdash B^{\perp}, \parr(\Delta^{\perp})}\,\parr}{\vdash B^{\perp}\parr\parr(\Delta^{\perp})}\,\parr}{\vdash \parr(\Gamma^{\perp}), A^{\perp} \otimes (B^{\perp}\parr\parr(\Delta^{\perp}))}\,\otimes}{\vdash \parr(\Gamma^{\perp})\parr(A^{\perp} \otimes (B^{\perp}\parr\parr(\Delta^{\perp})))}\,\parr$$

□

6. Generating the type of a formula-tree module

In this section we show (Theorem 22) that the type of a formula-tree module F is exactly the big union of the pretypes of all the reducts obtained by the iterative rewriting of the module F.

Proposition 19. *(Danos-Regnier [DR89]) Given a formula-tree module F then $P_F \subseteq O_{F^{\perp}}$.*

Proof. To see this, consider the following deductive system of (ordinary) partitions over a set containing the element 0 (we consider them as "pointed" partitions). For a partition, instead of $\{p_1, \ldots, p_m\}$ we write $\vdash p_1, \ldots, p_m$; $p = \{\Gamma_1\}, \ldots, \{\Gamma_m\}$ and $q = \{\Delta_1\}, \ldots, \{\Delta_{m'}\}$ are lists of classes; Γ, the Γ_k, Δ and the Δ_k lists of elements, all disjoint in the appropriate sense.

$$\dfrac{}{\vdash \{k, 0\}}\,\text{ax} \qquad \dfrac{\vdash p, \{\Gamma, 0\} \qquad \vdash q, \{\Delta, 0\}}{\vdash p, q, \{\Gamma, \Delta, 0\}}\,\otimes$$

$$\dfrac{\vdash p, \{\Gamma, 0\} \qquad \vdash q, \{\Delta, 0\}}{\vdash p, q, \{\Gamma, 0\}, \{\Delta\}}\,\parr\text{L} \qquad \dfrac{\vdash p, \{\Gamma, 0\} \qquad \vdash q, \{\Delta, 0\}}{\vdash p, q, \{\Gamma\}, \{\Delta, 0\}}\,\parr\text{R}$$

First observe that every switching s of the module F corresponds to a binary tree with nodes $\otimes$, $\parr$L and $\parr$R, whence to a proof tree π_s in this system. Moreover π_s derives a partition p such that after forgetting the element 0 (notation $\overline{p}$) we obtain exactly $p_{F,s}$, the partition induced by s.

We claim that for each switching s of the module F, inducing the partition $p_{F,s}$, there is a sequent calculus derivation of the orthogonal formula $F^\perp$ from the organization $p_{F,s}$ of its subformulas. Actually, we will prove by induction on the 'construction' s of p that there is a pointed sequent calculus derivation of $F^\perp$ from the (pointed) organization p: a derivation from organization $\overline{q}$ in which the hypothesis corresponding to the class of p containing 0 is marked.

The base case given by

$$\frac{}{\vdash \{k, 0\}}\,\text{ax} \quad \mapsto \quad \boxed{\vdash k}$$

If F is complex, let us assume that

$$\begin{array}{c}\pi\\ \vdash p, \{\Gamma, 0\}\end{array} \quad \mapsto \quad \frac{\vdash \Gamma_1 \quad \dots \quad \vdash \Gamma_m \quad \boxed{\vdash \Gamma}}{\vdash F_1^\perp}$$

and that

$$\begin{array}{c}\pi'\\ \vdash q, \{\Delta, 0\}\end{array} \quad \mapsto \quad \frac{\vdash \Delta_1 \quad \dots \quad \vdash \Delta_{m'} \quad \boxed{\vdash \Delta}}{\vdash F_2^\perp}$$

If the last rule deriving the partition of the module F is the $\otimes$-rule ($F = F_1 \otimes F_2$), we define

$$\frac{\begin{array}{c}\pi\\ \vdash p, \{\Gamma, 0\}\end{array} \qquad \begin{array}{c}\pi'\\ \vdash q, \{\Delta, 0\}\end{array}}{\vdash p, q, \{\Gamma, \Delta, 0\}}\,\otimes \quad \mapsto$$

$$\frac{\dfrac{\vdash \Delta_1 \quad \dots \quad \vdash \Delta_{m'} \qquad \dfrac{\vdash \Gamma_1 \quad \dots \quad \vdash \Gamma_m \quad \boxed{\vdash \Gamma, \Delta}}{\vdash F_1^\perp, \Delta}}{\vdash F_1^\perp, F_2^\perp}}{\vdash F_1^\perp \parr F_2^\perp}$$

which yields a pointed derivation of $F^\perp = F_1^\perp \parr F_2^\perp$ from organization $\big\{p, q, \{\Gamma, \Delta, 0\}\big\}$.

If the last rule deriving the partition of the module F is the $\parr$L-rule ($F = F_1 \parr F_2$), we define

$$\frac{\begin{array}{c}\pi\\ \vdash p, \{\Gamma, 0\}\end{array} \qquad \begin{array}{c}\pi'\\ \vdash q, \{\Delta, 0\}\end{array}}{\vdash p, q, \{\Gamma, 0\}, \{\Delta\}}\,\parr\text{L} \quad \mapsto$$

$$\frac{\dfrac{\vdash \Gamma_1 \quad \dots \quad \vdash \Gamma_m \quad \boxed{\vdash \Gamma}}{\vdash F_1^\perp} \qquad \dfrac{\vdash \Delta_1 \quad \dots \quad \vdash \Delta_{m'} \quad \vdash \Delta}{\vdash F_2^\perp}}{\vdash F_1^\perp \otimes F_2^\perp}$$

which yields a pointed derivation of $F^\perp = F_1^\perp \otimes F_2^\perp$ from organization $\big\{p, q, \{\Gamma, 0\}, \{\Delta\}\big\}$. The $\parr$R-rule is similar.

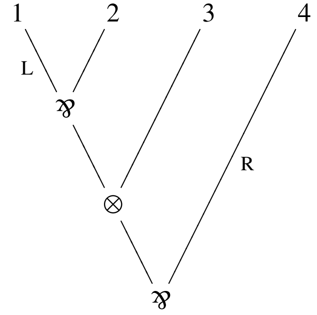
Observe that there is an alternative assignment for the $\otimes$-rule: if the last rule deriving the partition of the module $F = F_1 \otimes F_2$ is the $\otimes$-rule, we could also have defined

$$\begin{array}{c} \begin{array}{cc} \pi & \pi' \\ \vdash p, \{\Gamma, 0\} & \vdash q, \{\Delta, 0\} \end{array} \\ \hline \vdash p, q, \{\Gamma, \Delta, 0\} \end{array} \otimes \quad \mapsto$$

$$\begin{array}{c} \begin{array}{cc} \vdash \Gamma_1 \quad \ldots \quad \vdash \Gamma_m & \begin{array}{c} \vdash \Delta_1 \quad \ldots \quad \vdash \Delta_{m'} \quad \boxed{\vdash \Gamma, \Delta} \\ \hline\hline \vdash \Gamma, F_2^{\perp} \end{array} \end{array} \\ \hline\hline \vdash F_1^{\perp}, F_2^{\perp} \\ \hline \vdash F_1^{\perp} \parr F_2^{\perp} \end{array}$$

which yields a priori another pointed derivation of $F^{\perp} = F_1^{\perp} \parr F_2^{\perp}$ from organization $\big\{p, q, \{\Gamma, \Delta, 0\}\big\}$.

As an example, the switching



(0)

induces the element $\big\{\{1, 3\}, \{2\}, \{4\}\big\}$ of the pretype of $((1 \parr 2) \otimes 3) \parr 4$. We derive this partition by the derivation as indicated below on the left, and it translates into the derivation of $((1 \otimes 2) \parr 3) \otimes 4$ on the right.

$$\begin{array}{c} \begin{array}{cc} \begin{array}{c} \begin{array}{cc} \begin{array}{c} \begin{array}{cc} \begin{array}{c} \hline \vdash \{1, 0\} \end{array} & \begin{array}{c} \hline \vdash \{2, 0\} \end{array} \end{array} \\ \hline \vdash \{1, 0\}, \{2\} \end{array} \parr\text{L} & \begin{array}{c} \hline \vdash \{3, 0\} \end{array} \end{array} \\ \hline \vdash \{1, 3, 0\}, \{2\} \end{array} \otimes & \begin{array}{c} \hline \vdash \{4, 0\} \end{array} \end{array} \\ \hline \vdash \{1, 3\}, \{2\}, \{4, 0\} \end{array} \parr\text{R} \quad \mapsto$$

$$\begin{array}{c} \begin{array}{cc} \begin{array}{c} \begin{array}{c} \vdash 1, 3 \qquad \vdash 2 \\ \hline \vdash 1 \otimes 2, 3 \end{array} \\ \hline \vdash (1 \otimes 2) \parr 3 \end{array} & \vdash 4 \end{array} \\ \hline \vdash ((1 \otimes 2) \parr 3) \otimes 4 \end{array}$$

In general the inclusion $P_F \subseteq O_{F^{\perp}}$ is strict. We borrow the example from [DR89], where F is $((1 \parr 2) \otimes (3 \parr 4)) \otimes (5 \otimes 6)$. Then for every $p \in P_F$ the elements 5 and 6 will be in the same class, while $O_{F^{\perp}}$ contains $\big\{\{1, 3\}, \{2, 5\}, \{4, 6\}\big\}$:

$$\begin{array}{c} \begin{array}{cc} \begin{array}{c} \vdash 1, 3 \qquad \vdash 2, 5 \\ \hline \vdash 1 \otimes 2, 3, 5 \end{array} & \vdash 4, 6 \end{array} \\ \hline \vdash 1 \otimes 2, 3 \otimes 4, 5, 6 \\ \hline \vdash ((1 \otimes 2) \parr (3 \otimes 4)), (5 \parr 6) \\ \hline \vdash ((1 \otimes 2) \parr (3 \otimes 4)) \parr (5 \parr 6) \end{array}$$

One could ask if there is an inductive definition of $O_{F^\perp}$. For $F^\perp = F_1^\perp \otimes F_2^\perp$ we can show that $O_{F^\perp}$ is the pointwise union of $O_{F_1^\perp}$ and $O_{F_2^\perp}$. However, if $F^\perp = F_1^\perp \parr F_2^\perp$ then $O_{F^\perp}$ is not the pointwise 'merge' of $O_{F_1^\perp}$ and $O_{F_2^\perp}$. Indeed, again the above example shows that in any such merge 5 and 6 are in the same class, since $O_{5 \parr 6} = \big\{\{5, 6\}\big\}$.

Lemma 20. *If $F \to^* F'$ then $P_{F'} \subseteq O_{F^\perp}$.*

Proof. $P_{F'} \subseteq O_{F'^\perp}$ by Proposition 19 and $O_{F'^\perp} \subseteq O_{F^\perp}$ by Lemma 17. □

The next theorem shows that every organization O_F is a fact.

Theorem 21. *(Danos-Regnier [DR89]) Let $F = F(\alpha_{i_1}, \dots, \alpha_{i_n})$ be a formula-tree with n different atomic subformulas. Then*

$$(P_F)^\perp = O_F.$$

Proof. Given $p \in O_F$ we build a sequent calculus derivation of F from generalized axioms according to organization p. The corresponding proof-net shows that $\{p\} \perp P_F$.

The other way around, given a proof-net $\hat{p} \amalg F$, sequentialization gives a proof of F from the generalized axioms, showing $p \in O_F$. □

Theorem 22 (Type of a formula-tree). *Given a formula-tree module F, then*

$$T_F = \bigcup_{F':F \to^* F'} P_{F'}.$$

Proof. By $P_F^{\perp\perp} = P_{F^\perp}^\perp = O_{F^\perp}$ it is enough to show

$$O_{F^\perp} \subseteq \bigcup_{F':F \to^* F'} P_{F'}$$

since the other inclusion holds by Lemma 11 for arbitrary modules. We apply induction on the size of F.

If F is an atom the result is clear.

If $F = G \parr H$, then

$$\begin{aligned} O_{F^\perp} &= O_{G^\perp \otimes H^\perp} \\ &= \big\{ g \cup h \mid g \in O_{G^\perp}, h \in O_{H^\perp} \big\} \\ &\subseteq \left\{ g \cup h \mid g \in \bigcup_{G':G \to^* G'} P_{G'}, h \in \bigcup_{H':H \to^* H'} P_{H'} \right\} \\ &\subseteq \bigcup_{F':F \to^* F'} P_{F'}. \end{aligned}$$

If $F = G \otimes H$ we apply induction on the complexity $c(F)$. If $c(F) = 0$ then F is a bipole, hence a $\otimes$-tree, and the result is immediate. If $c(F) > 0$ then by

Lemma 18 given $p \in O_{F^\perp}$ we know there is an F' reduct of F of strictly lower complexity such that $p \in O_{F'^\perp}$, hence by induction hypothesis

$$p \in \bigcup_{F'':F' \to^* F''} P_{F''} \subseteq \bigcup_{F'':F \to^* F''} P_{F''}.$$

□

7. Generating the type of a general module

Theorem 22 says that the iteration of the rewrite relation → allows us to generate the full type of a formula-tree module. But does Theorem 22 generalize to all modules, with possibly, axiom links? In other words, does the following equality

$$P_M^{\perp\perp} = \bigcup_{M':M \to^* M'} P_{M'}$$

hold when M is an arbitrary module? Unfortunately the answer is “no”, as we will see.

Let M be an arbitrary correct cut-free module with one proper conclusion and with border consisting entirely of hypotheses. We write $M = F \amalg A$ where F is a formula and A a set of axiom-links. Now

$$P_M = \{\overline{p} \mid p \in P_F\}$$

where $\overline{p}$ is obtained from p by contracting the classes according to A. We could ask the question whether the same holds for the types:

$$P_M^{\perp\perp} = \left\{\overline{p} \mid p \in P_F^{\perp\perp}\right\} \tag{3}$$

If yes, then the result of Theorem 22

$$P_F^{\perp\perp} = \bigcup_{F':F \to^* F'} P_{F'}$$

would generalize to arbitrary modules:

$$\begin{aligned} P_M^{\perp\perp} &= \left\{\overline{p} \mid p \in P_F^{\perp\perp}\right\} \\ &= \left\{\overline{p} \mid p \in \bigcup_{F':F \to^* F'} P_{F'}\right\} \\ &= \bigcup_{F':F \to^* F'} \{\overline{p} \mid p \in P_{F'}\} \\ &= \bigcup_{M':M \to^* M'} P_{M'}. \end{aligned}$$

However, we only know:

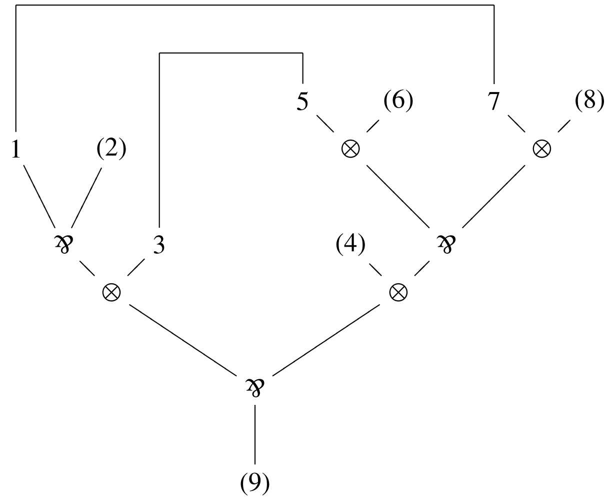
$$P_M = \{\overline{p} \mid p \in P_F\} \subseteq \left\{\overline{p} \mid p \in P_F^{\perp\perp}\right\} \subseteq P_M^{\perp\perp} \tag{4}$$

where the last inclusion is a consequence of the fact that for every $q \in P_M^{\perp}$ the partition $q \cup a$ belongs to $P_F^{\perp}$. Consequently (3) holds if either P_M is a fact or M is a formula-tree (so, there are no axioms at all) or M has empty border (i.e. is a proof-net). But in general the last inclusion of (4) is strict, even if P_F is a fact, as we will see in the following example.

Example 23. Let F be the formula $((1 \parr 2) \otimes 3) \parr (4 \otimes ((5 \otimes 6) \parr (7 \otimes 8)))$. Its pretype equals

$$\begin{aligned} P_F = \{\ &\{\{1\},\{2,3\},\{5,6\},\{4,7,8\}\}\ ,\ \{\{1\},\{2,3\},\{7,8\},\{4,5,6\}\}\ , \\ &\{\{2\},\{1,3\},\{5,6\},\{4,7,8\}\}\ ,\ \{\{2\},\{1,3\},\{7,8\},\{4,5,6\}\}\ \} \end{aligned}$$

and we can easily check that its type coincides with its pretype, so P_F is actually a fact (e.g, by Theorem 22). Now, by attaching the axiom-links $\{1, 7\}$ and $\{3, 5\}$ we get the next module M



whose pretype is

$$P_M = \{\ \{\{2\},\{4,6,8\}\}\ ,\ \{\{8\},\{2,4,6\}\}\ ,\ \{\{2,6\},\{4,8\}\}\ \}$$

Any partition $p \in P_M^{\perp}$ should consist of 3 classes, hence 2 singletons and one pair. The element 4 cannot be in the same class as 6 or 8 or 2. Also 6 cannot be in the same class with 4 or 8 or 2. So there is only one possibility for p, namely $p = \{\{4\},\{6\},\{2,8\}\}$, which is indeed in $P_M^{\perp}$:

$$P_M^{\perp} = \{\ \{\{4\},\{6\},\{2,8\}\}\ \}.$$

Its complement consists of partitions with 2 classes of the form $\{\{2,\dots\},\{8,\dots\}\}$, whence

$$P_M^{\perp\perp} = \{\ \{\{2\},\{8,4,6\}\}\ ,\ \{\{2,6\},\{8,4\}\}\ ,\ \{\{2,4\},\{8,6\}\}\ ,\ \{\{2,4,6\},\{8\}\}\ \}$$

which strictly contains

$$\left\{\overline{p} \mid p \in P_F^{\perp\perp}\right\}$$

because the latter is equal to $\{\overline{p} \mid p \in P_F\}$ (P_F is a fact), i.e. to P_M.

Why is $q := \{\{2,4\},\{8,6\}\}$ not in $\left\{\overline{p} \mid p \in P_F^{\perp\perp}\right\}$? Let us examen all partitions p over the border $\{1,2,3,4,5,6,7,8\}$ such that $\overline{p} = q$. We obtain them by breaking, successively for every axiom link $\{a,b\}$, any class X of q in two parts, with a and b in the distinct new classes. E.g. for the axiom link $\{1,7\}$ we can break the class $\{2,4\}$ in four ways into two classes $\{1,\dots\},\{7,\dots\}$, yielding the partitions:

$$\begin{gathered}\{\ \{\{1\},\{7,2,4\},\{8,6\}\}\ ,\ \{\{1,4\},\{7,2\},\{8,6\}\}\ ,\\ \{\{1,2\},\{7,4\},\{8,6\}\}\ ,\ \{\{1,2,4\},\{7\},\{8,6\}\}\ \}\end{gathered}$$

and similar partitions when we break the class $\{8,6\}$ by axiom link $\{1,7\}$. Now, for the partition $\{\{1\},\{7,2,4\},\{8,6\}\}$ there are $2^1 + 2^3 + 2^2$ possibilities to break a class by axiom link $\{3,5\}$. We conclude that there are 104 candidate p satisfying $\overline{p} = q$.

So the question becomes: why does none of these candidate p belong to $P_F^{\perp\perp}$? The answer follows by a careful analysis of $(P_{F^\perp})^\perp$, which we know to be equal to $P_F^{\perp\perp}$ by Theorem 7. Let p be one of the above candidates and suppose $p \in (P_{F^\perp})^\perp$. This leads to a contradiction by the following reasoning.

If $\{2,4,\dots\} \in p$ or $\{6,8,\dots\} \in p$, then $p \notin (P_{F^\perp})^\perp$, which follows from appropriate switchings in the formula-tree $F^\perp$ that would yield a cycle. So both classes of q have to be nontrivially broken (in the above sense), i.e. p must be of the form $\{\{2,a\},\{4,b\},\{6,c\},\{8,d\}\}$ where $\{\{a,b\},\{c,d\}\} = \{\{1,7\},\{3,5\}\}$. In case $\{c,d\} = \{1,7\}$, then either $\{6,1\},\{8,7\} \in p$ or $\{6,7\},\{8,1\} \in p$. However, $\{6,1\} \notin p$ and $\{8,1\} \notin p$ by appropriate switchings. In case $\{c,d\} = \{3,5\}$, then either $\{6,3\},\{8,5\} \in p$ or $\{6,5\},\{8,3\} \in p$. However, $\{6,3\} \notin p$ and $\{8,3\} \notin p$.

Example 23 shows that in general the rewriting method is not able to extract the type of a module with axioms. In other words the rewrite method is not complete w.r.t. the types of all modules. However, this method does give a positive answer to the question of building the orthogonal of a general module (a straight consequence of the following Theorem 24). This is enough for building the type of any module.

Theorem 24 (Type orthogonal). *Let P be a set of partitions of a set X, then*

$$P^\perp = \bigcap_{p \in P} T_{\hat{p}^\perp}.$$

Proof. First observe that we can associate to any partition p its corresponding bipole $\hat{p}$ (see Example 1) whose pretype is then

$$\{p\} = P_{\hat{p}} = (P_{\hat{p}})^{\perp\perp} = (P_{\hat{p}^{\perp}})^{\perp} \tag{5}$$

by, respectively Example 1, Lemma 4 and Theorem 7. So, the orthogonal type of a set of partitions P is equal to

$$P^{\perp} = \bigcap_{p \in P} \{p\}^{\perp} = \bigcap_{p \in P} ((P_{\hat{p}^{\perp}})^{\perp})^{\perp} = \bigcap_{p \in P} T_{\hat{p}^{\perp}} \tag{6}$$

by Lemma 3 and (5). □

Theorem 24 expresses $P^{\perp}$ in terms of $T_{\hat{p}^{\perp}}$, which we know how to build by Theorem 22 ($\hat{p}^{\perp}$ being a formula-tree). In particular, for a module M we obtain T_M (i.e. $P_M^{\perp\perp}$) by applying Theorem 24 twice (once for P_M and once for $P_M^{\perp}$).

8. Conclusions, related and further works

We presented an algorithmic characterization of the type of a module which relies only on some geometrical properties of proof-nets. In particular, this procedure does not refer to the sequent calculus neither to an abstract notion of orthogonality. The procedure iterates inside the module some elementary steps of graph rewriting, illustrated in Section 4, and only based on the associativity and commutativity of $\otimes$ and $\parr$ and the weak-distributivity laws. These laws, corresponding to the theorems of linear logic $A \otimes (B \parr C) \vdash (A \otimes B) \parr C)$ and $A \otimes (B \parr C) \vdash (A \otimes C) \parr B$, are also been ivestigated in some papers on category theory, [CS97], and game semantics, [AJ94].

In [CS97] Cockett and Seely showed that the weak (or linear) distributivity is precisely what is needed to model Gentzen's cut rule (in the absence of other structural rules) and can be strengthened in a natural way to generate *-autonomous categories.

In Abramsky and Jagadeesan's paper on game semantics for linear logic, [AJ94] the following "rewrite" laws are mentioned[2]:

law 1 : let $\Gamma = C[A \otimes (B \parr C)]$, $\Gamma_1 = C[(A \otimes B) \parr C]$ and $\Gamma_2 = [(A \otimes C) \parr B]$ (binary) sequents[3], then:

$$(\forall i)\ \Gamma \vdash \Gamma_i \qquad \text{and} \qquad \vdash \Gamma \Leftrightarrow (\forall i)\ \vdash \Gamma_i$$

law 2 : let $\Gamma = C[A \otimes (B \otimes C)]$, $\Gamma_1 = C[A \otimes (B \parr C)]$ and $\Gamma_2 = C[A \parr (B \otimes C)]$ (binary) sequent, then:

$$(\forall i)\ \Gamma \vdash \Gamma_i \qquad \text{and} \qquad \vdash \Gamma \Leftrightarrow (\forall i)\ \vdash \Gamma_i$$

[2] These laws corresponds to Lemma 2 and 3 of the preliminary version of [AJ94] appeared in the Proceedings of the Conference on Foundations of Software Technology and Theoretical Computer Science, December 1992. The authors thank the anonymous referee for the reference to Proposition 9.

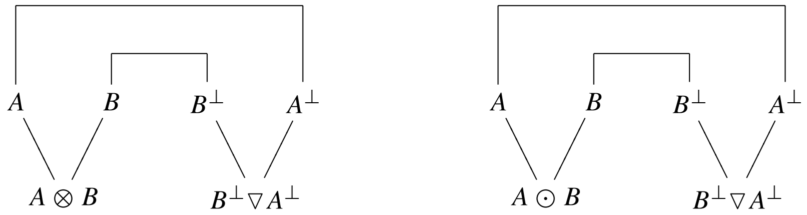
[3] Naively a binary sequent is a sequent where each pair of literals is specified with distinct atoms.

where $C[.]$ is a monotone context, i.e., with the hole [.] appearing only under the scope of tensors and par. Now, while the first law corresponds, clearly, to the linear distributive one, it is not so immediate the meaning of law 2 in terms of module rewrite[4].

As further work we aim to extend our rewrite method to other fragments of linear logic like the non-commutative one (NL, [AR00]). Actually, due to some intrinsic difficulties of the NL sequent calculus, it seems quite hard to try to extend to such a fragment techniques like the so called method of the "organization of a formula-tree" (seen in section 5), for getting the type of a module. Naively, the Danos-Regnier method would identify the organizations of the two tensors of NL, $\otimes$ (commutative) and $\odot$ (non commutative):

$$O(A \otimes B) = O(A \odot B) = \{\{\{A\}, \{B\}\}\}.$$

This would mean the two modules $M' = A \otimes B$ and $M'' = A \odot B$ have the same type, so they behave in the same way when they are plugged to the same (non-commutative) module, which is, in general, not true. If we branch M', respectively M'' with the module $M = B^{\perp} \nabla A^{\perp}$ we get only one correct proof-net of MNL, the one on the right hand side:



A naive solution could be to add some structure (order) to the partitions belonging to the type of a non-commutative module. This goal could be reached by: *(i)* finding the good rewrite rules for the non-commutative structures, *(ii)* extracting from the final objects of the rewriting (the non-commutative bipoles), the good structure for those partitions corresponding to the bipoles.

Moreover, the rewrite method could also implement a (distributed) theorem prover based on proof-nets. E.g., assume we want to find a proof-net with the unique conclusion F (we can always assume a proof-net with a unique conclusion, by a terminal generalized ⅋-link): first we build both the formula-trees of F and $F^{\perp}$, then we start with rewriting $F^{\perp}$ until we get the first bipole with only binary $\otimes$-trees. If F is provable then there will always exist at least such a bipole with only binary $\otimes$-subtrees: these binary $\otimes$-trees play the role of axioms for the module F.

[4] At the moment we are investigating this law.

References

[AJ94] Abramsky, S., Jagadeesan, R.: Games and Full Completeness for Multiplicative Linear Logic. J. Symbolic Logic **59** (2), 543–574 (1994)

[AR00] Abrusci, V.M., Ruet, P.: Non commutative logic I: the multiplicative fragment. Annals of Pure and Applied Logic **101**, 29–64 (2000)

[And92] Andreoli, J.-M.: Logic programming with focusing proofs in linear logic. Journal of Logic and Computation, **2**, 1992

[And01] Andreoli, J.-M.: Focussing and proof construction. Annals of Pure and Applied Logic **107**, 131–163 (2001)

[CS97] Cockett, J.R.B., Seely, R.A.G.: Weakly Distributive Categories. Journal of Pure and Applied Algebra, **114**, 133–173 (1997)

[DR89] Danos, V., Regnier, L.: The structure of multiplicatives. Archive for Mathematical Logic **28**, 181–203 (1989)

[Gir87] Girard, J.-Y.: Linear logic. Theoretical Computer Science **50**, 1–102 (1987)

[Gir87a] Girard, J.-Y.: Multiplicatives. Rendiconti del Seminario Matematico dell'Università e Policlinico di Torino, special issue on Logic and Computer Science, pp. 11–33, 1987